

1  
2 **CLAIMS**

3 1. A method, comprising:  
4 receiving an object, the object being structured according to a first protocol;  
5 evaluating at least a portion of the object against one or more queries  
6 structured according to a second protocol, the second protocol being different than  
7 the first protocol; and  
8 wherein the evaluating step is performed without serializing the object to  
9 reconstruct the object according to the second protocol.  
10

11 2. The method as recited in claim 1, wherein:  
12 the first protocol further comprises Common Language Runtime (CLR)  
13 protocol; and  
14 the second protocol further comprises eXtensible Markup Language (XML)  
15 protocol.  
16

17 3. The method as recited in claim 1, wherein the evaluating step further  
18 comprises:  
19 accessing an info set that maps to the object;  
20 locating a value being evaluated by the query using the info set; and  
21 evaluating the query according to at least the located value.  
22

23 4. The method as recited in claim 3, wherein the accessing the info set  
24 step further comprises retrieving the info set from memory.  
25

1  
2       5.     The method as recited in claim 3, further comprising constructing at  
3 least a portion of the infoaset.

4  
5       6.     The method as recited in claim 3, wherein the accessing the infoaset  
6 step further comprises retrieving a portion of the infoaset from memory, and the  
7 method further comprises constructing at least an additional portion of the infoaset.

8  
9       7.     The method as recited in claim 3, further comprising storing the  
10 infoaset for re-use when the object is encountered in another input.

11  
12       8.     The method as recited in claim 1, wherein the evaluating step further  
13 comprises:

14       creating a template according to the second protocol;

15       locating a template element that corresponds with a query value;

16       locating an object property associated with the template element; and

17       evaluating a value of the object property against the query value.

18  
19       9.     The method as recited in claim 8, further comprising the step of  
20 generating one or more opcodes that initiate the evaluating step when the object is  
21 encountered in a subsequent query evaluation process.

1           **10.**     The method as recited in claim 1, further comprising:  
2           generating one or more opcodes that, when executed, perform at least a  
3           portion of the evaluating step;  
4           compiling and executing the one or more opcodes at runtime; and  
5           wherein executing the one or more compiled opcodes retrieves property  
6           values and fields directly from the object for evaluation.

7  
8           **11.**     A system, comprising:  
9           one or more filters, each filter being structure according to a first protocol;  
10          an input module configured to receive an input for evaluation against at  
11          least a query, the input including an object that is structured according to a second  
12          protocol that is different from the first protocol;  
13          a mapping module configured to map one or more properties of the object  
14          to a template arranged according to the first protocol without serializing the object  
15          or any portion thereof; and  
16          a filter engine configured to locate a query value in the object by referring  
17          to the template to locate an object value corresponding to the query value, to  
18          evaluate the query value against the object value, and to return a result of the  
19          evaluation.

20  
21          **12.**     The system as recited in claim 11, wherein the mapping module is  
22          further configured to utilize an infoset model corresponding to the object to map  
23          the object properties to the template.  
24  
25

1           **13.**     The system as recited in claim 12, wherein the mapping model is  
2 further configured to retrieve at least a portion of the infoaset model from memory.

3  
4           **14.**     The system as recited in claim 12, wherein the mapping module is  
5 further configured to construct at least a portion of the infoaset.

6  
7           **15.**     The system as recited in claim 11, further comprising an opcode  
8 generation module configured to generate one or more opcodes that, when  
9 executed, perform one or more steps taken in the process of evaluating the object  
10 so that the one or more opcodes can be used in subsequent evaluations regarding  
11 the object to automatically perform the one or more steps.

12  
13           **16.**     The system as recited in claim 15, wherein the opcodes are compiled  
14 and executed at runtime.

15  
16           **17.**     The system as recited in claim 11, wherein the query is an  
17 eXtensible Markup Language (XML) query and the object is a Common Runtime  
18 Language (CLR) object.

19  
20           **18.**     One or more computer-readable media comprising computer-  
21 executable instructions that, when executed on a computer, perform the following  
22 steps:

23           creating an object model that maps object properties to an object template  
24 that conforms to a query protocol, wherein the object conforms to an object  
25 protocol that is different than the query protocol;

1 identifying a query value;  
2 referencing the object template to locate an object property corresponding  
3 to the query value;  
4 identifying a property value assigned to the object property; and  
5 evaluating the property value against the query value to determine if the  
6 query is satisfied by the property value.  
7

8 **19.** The one or more computer-readable media as recited in claim 18,  
9 wherein the object properties are mapped to the object template and the property  
10 value is evaluated against the query without serializing object data.  
11

12 **20.** The one or more computer-readable media as recited in claim 18,  
13 wherein the query protocol is eXtensible Markup Language (XML) and the object  
14 protocol is Common Language Runtime (CLR).  
15

16 **21.** The one or more computer-readable media as recited in claim 18,  
17 wherein the object model is an infoset model that corresponds to the object.  
18

19 **22.** The one or more computer-readable media as recited in claim 18,  
20 wherein the creating an object model further comprises creating only a portion of  
21 the object model that is necessary to discover the object property tested by the  
22 query.  
23  
24  
25

1           **23.**     The one or more computer-readable media as recited in claim 18,  
2 further comprising storing the object model so that it can be retrieved for future  
3 query evaluations against the object to avoid having to re-create the object model.

4  
5           **24.**     The one or more computer-readable media as recited in claim 18,  
6 wherein the creating an object model further comprises retrieving a partially  
7 completed version of an object model and augmenting the object model to an  
8 extent necessary to locate the object property that corresponds to the query value.

9  
10          **25.**     The one or more computer-readable media as recited in claim 18,  
11 further comprising:  
12           generating one or more opcodes to perform the referencing step, the  
13 identifying a property value step and the evaluating step;  
14           storing the one or more opcodes in memory; and  
15           wherein the opcodes can be retrieved and utilized to perform similar steps  
16 in a subsequent query evaluation involving the object.

17  
18          **26.**     The one or more computer-readable media as recited in claim 25,  
19 wherein the opcodes can be compiled and executed dynamically at runtime.

20  
21          **27.**     The one or more computer-readable media as recited in claim 18,  
22 wherein the query further comprises an XPath filter.

1  
2       **28.**     A filter engine stored on one or more computer-readable media,  
3 comprising:  
4       one or more filters derived from a query language, each filter containing at  
5 least one condition specifying a query value that can be compared to one or more  
6 filter engine inputs;  
7       an input function configured to receive an input for comparison against one  
8 or more of the filters wherein the input may contain an object derived from an  
9 object language that is different from the query language;  
10      a mapping module configured to reference an object info set and map at  
11 least one object property identified in the object info set to an object template that  
12 is in accordance with the query language without serializing data included in the  
13 object; and  
14      wherein the filter engine is further configured to locate an object property  
15 value by referencing the object template to locate the object property in the object  
16 info set, and to evaluate the object property value against the query value.

17  
18       **29.**     The filter engine as recited in claim 28, wherein the mapping  
19 module is further configured to create the object info set or to augment a partial  
20 object info set retrieved from memory.

21  
22       **30.**     The filter engine as recited in claim 28, wherein the mapping  
23 module is further configured to store the object info set if the mapping module has  
24 changed the object info set.  
25

1           **31.**     The filter engine as recited in claim 28, further comprising:  
2           an opcode generation module configured to generate one or more opcodes  
3     to automatically perform one or more of the filter engine procedures upon a first  
4     encounter with the object;  
5           an opcode store for storing opcodes generated by the opcode generation  
6     module; and  
7           wherein the one or more opcodes can be used to perform the one or more  
8     filter engine procedures upon a subsequent encounter with the object.

9  
10           **32.**     The filter engine as recited in claim 31, wherein the opcodes can be  
11     dynamically compiled and executed at runtime.

12  
13           **33.**     The filter engine as recited in claim 28, wherein the query language  
14     is XPath and the object language is a Common Language Runtime (CLR)  
15     language.

16  
17           **34.**     The filter engine as recited in claim 28, wherein the query language  
18     and the object language are hierarchically structured so that hierarchy levels of the  
19     object info set have corresponding level in the hierarchy of the object template.



1  
2       **35.**     A method, comprising:  
3       mapping object properties to template elements;  
4       identifying a query value in a query against which the object is to be  
5 evaluated;  
6       referencing the template to identify an element corresponding to the query  
7 value;  
8       identifying an object property value corresponding to the identified  
9 template element;  
10      comparing the object property value to the query value to evaluate at least a  
11 portion of the query; and  
12      wherein the object is derived from an object language, the query is derived  
13 from a query language, and the steps are accomplished without serializing data  
14 included with or referenced by the object.

15  
16       **36.**     The method as recited in claim 35, wherein the object language  
17 further comprises a Common Language Runtime (CLR) language.

18  
19       **37.**     The method as recited in claim 35, wherein the query language  
20 further comprises eXtensible Markup Language (XML).

21  
22       **38.**     The method as recited in claim 35, wherein the query language is  
23 XPath.  
24  
25

1  
2       **39.**     The method as recited in claim 35, further comprising using an  
3 object infoaset model that references object properties to map the object properties  
4 to template elements.  
5

6       **40.**     The method as recited in claim 39, further comprising building the  
7 infoaset model at least to the extent necessary to identify the correct object property  
8 value.  
9

10       **41.**    The method as recited in claim 39, further comprising retrieving the  
11 infoaset model from memory.  
12

13       **42.**    The method as recited in claim 39, further comprising:  
14       retrieving a partial infoaset model from memory; and  
15       if the object property value cannot be identified from the partial infoaset,  
16 augmenting the infoaset model at least to the extent necessary to identify the correct  
17 object property value.  
18  
19  
20  
21  
22  
23  
24  
25